HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 70006393 -1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Lakshmi Kutty Cheeniyil et al

Confirmation No.: 9525

Application No.: 09/943027

Examiner: Thai, Cang G

Filing Date: Aug 29, 2001

Group Art Unit: 3629

Title: Migration Of A Workflow System To Changed Process Definitions

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on July 16, 2006.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

□ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

□ 1st Month $120 □ 2nd Month $450 □ 3rd Month $1020 □ 4th Month $1590

□ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of ___________. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

Respectfully submitted,

Lakshmi Kutty Cheeniyil et al

By ______________________

Thi Dang

Attorney/Agent for Applicant(s)

Reg No. : 53,877

Date : September 11, 2006

Telephone : +65 6374 6179 (Singapore)

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 70006393 -1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Lakshmi Kutty Cheeniyil et al	Confirmation No.:	9525
Application No.:	09/943027	Examiner:	Thai, Cang G
Filing Date:	Aug 29, 2001	Group Art Unit:	3629

Title: Migration Of A Workflow System To Changed Process Definitions

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on July 16, 2006.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

□ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

□ 1st Month $120	□ 2nd Month $450	□ 3rd Month $1020	□ 4th Month $1590

□ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of ___________. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

Respectfully submitted,

Lakshmi Kutty Cheeniyil et al

By

Thi Dang

Attorney/Agent for Applicant(s)

Reg No. :	53,877
Date :	September 11, 2006
Telephone :	+65 6374 6179 (Singapore)

Table of Contents

I. Real Party in Interest

The present application is assigned to Hewlett Packard Development Company L.P.

II. Related Appeals and Interferences

The Appellant's legal representative, or assignee, does not know of any other appeal or interferences which will affect or be directly affected by or have bearing on the Board's decision in the pending appeal.

III. Status of Claims

Claims 1-10 are pending in the application and are being appealed.

IV. Status of Amendments

Amendments filed on June 9, 2005 have been entered and acted upon by the Examiner.

No amendment after Final rejection has been filed.

V. Summary of Claimed Subject Matter

Appellant's specification discloses a method for executing a work flow in a WFMS (workflow management system) having at least one process instance executing an original process definition, and migrating the process instance to a changed process definition. Claim 1-9 cover an embodiment of the method which includes checking each process instance during the execution of the original process definition whether the process instance meets a migration condition (page 5, lines 28-31), and migrating each process instance during the execution of the original process definition to a modified process definition if the migration condition is met such that the process instance executes the changed process definition (page 6, lines 27-33, page 7, lines 2-6).

The checking of each process instance includes defining a set of worst case migration points (WMP), and migrating the process instance to the modified process definition if its execution has not gone beyond any of the WMP (page 6, lines 14-34). The set of the WMP may be defined by reading from an user input, or by computing the set of WMP based on the original and modified process definitions (page 8, lines 5-15).

The specification also discloses a method for creating a process definition to be executed by a WFMS. Claim 10 covers an embodiment of the method which includes defining an original process definition to be executed in a workflow system, starting execution of the process instance as per the original process definition, defining a modified process definition, checking for each process instance whether a migration condition is met, and replacing the nodes of the original process definition in a running process instance with corresponding nodes of the modified process definition. Only the nodes of the original process definition which satisfy the migration condition are replaced with the corresponding nodes of the modified process definition. After the nodes are replaced, the running process instance executes the modified process definition (page 4 lines 26-29, page 5 lines 5-31, page 6 lines 1-34, page 7 lines 2-5).

VI. Grounds of Rejection to be Reviewed on Appeal

Whether Claims 1-10 are patentable over Underwood (US Patent No. 6,704,873) under 35 U.S.C. §102(e)?

VII. Argument

The 35 U.S.C. 102 rejection based on Underwood

Claims 1-10 were rejected under 35 U.S.C. 102(e) as being anticipated by Underwood.

A. Claim 1

Appellant's Claim 1 recites a method for executing a work flow in a WFMS having at least one process instance executing an original process definition and migrating the process instance to a changed process definition, comprising:

"a) checking each process instance during the execution of the original process definition whether the process instance meets a migration condition; and

b) migrating each process instance during the execution of the original process definition to a modified process definition if the migration condition is met such that said process instance executes said changed process definition."

The invention can be advantageously applied when long-running instances of processes according to a process definition are executed and the process definition has to be changed, e.g., due to external reasons. The invention provides a method that allows migration of the running process instances such that after migrating, the running process instances conform to the changed process definition.

Underwood Disclosure

Underwood describes a system and method of providing a global internetworking gateway architecture in an e-commerce environment. In particular, a method for providing an activity framework is described. According to this method, a plurality of sub-activities are created, wherein each includes sub-activity logic adapted to generate output based on an input received from a user upon execution. Further, a plurality of activities are defined, each of which executes the sub-activities in a unique manner upon being selected for accomplishing a goal associated with the activity.

Examiner's Arguments

With respect to independent Claim 1, the Examiner asserted that Underwood discloses:

"a method for executing a work flow in a WFMS having at least one process instance executing an original process definition, and migrating the said process instance to changed process definition, said method comprising the following steps:

a) checking each process instance during the execution of the original process definition whether the process instance meets a migration condition [checking the original process to determine if it meets the condition for data transfer] {Column 28, Lines 45-58 wherein this reads over "first, in operation 800 a plurality of sub-activities are created which each include sub-activity logic adapted to generate output based on a input received from a user upon execution"}; and

b) migrating each process instance during the execution of the original process definition to a modified process definition if the migration condition is met such that said process instance executes said changed process definition [modifying the data transfer once the condition is met during the execution] {Column 28, Lines 48-51, wherein this reads over "in operation 802, a plurality of activities are defined, each of which execute the sub-activities in a unique manner upon being selected for accomplishing a goal associated with the activity"}."

The Examiner's rejection of Claim 1 under 35 U.S.C. 102(e) over Underwood is premised on his assertion that the generation of the output based on the input received from a user corresponds to checking whether the process instance meets a migration condition as recited in Claim 1. In addition, the Examiner asserted that the execution of the sub-activities in a unique manner upon being selected for accomplishing an associated goal corresponds to migrating each process instance to the modified process definition.

The Final Office Action, mailed on April 18, 2006, also responded to Appellant's arguments by stating that

"*Underwood teaches the aspects of the process instance in the migration control tools that control multiple versions of source code, data, and other items as they are changed, tested, and moved from one development environment into another, for example, from development to test and from test to production (Column 78, Lines 41-45)*" and "*the source code is required in order for execution to take place, for example for each code fix, a complete assembly test may be re-executed (Column 191, Lines 66-67)*".

Appellant's Reply

Underwood does not disclose that a process instance is checked during the execution of an original process definition whether the process instance meets a migration condition and that the process instance is migrated to a modified process definition such that said process instance, after migrating, executes the changed process definition as recited in Claim 1.

The generation of output based on an input and the corresponding processing of the input data is not related to checking whether a process meets a migration condition, i.e. is suitable to be migrated to a modified process definition. In particular, nothing is being said about a condition being checked by the sub-activities.

Further, the execution of the sub-activities by the activities in a unique way does not imply that a definition of a process is changed and that a process instance is migrated to the changed process definition after checking that the migration condition has been met.

In particular, it is not disclosed in Underwood that the definition of the sub-activities changes. Also, it is not disclosed that the way of generation of output based on an input changes. Additionally, it is not said in Underwood that the execution of the sub-activities by the activities is done depending on the result of the check on any migration condition. Therefore, the method and system described in Underwood is not related to the migrating of a process instance from an original process definition to a modified process definition as recited in Claim 1.

In addition, Appellant submits that the Migration Control Tool mentioned in Underwood (column 78 lines 41-45) is a software control tool that handles multiple versions of a software program. This software control tool is completely different from the workflow system which allows the migration of process instances from the original definition to the modified definition. Further, there is no mention that a process instance of a workflow system is checked during the execution of an original process definition whether the process instance meets a migration condition as recited in Claim 1. The requirement of the source code for execution to take place as stated in the Final Office Action does not relate to checking whether the process instance of a workflow system meets the migration condition as recited in Claim 1.

As Underwood fails to disclose each and every element contained in Claim 1, Underwood cannot anticipate Claim 1. Accordingly, Claim 1 is patentable over Underwood. Claims 2-9 are also patentable over Underwood at least by virtue of their dependency on Claim 1.

B. Claim 10

Appellant's Claim 10 recites a method for creating a process definition to be executed by a WFMS. The method comprises:

> "a) defining an original process definition to be executed in a workflow system,
> b) starting execution of the process instance as per the original process definition,
> c) defining a modified process definition,
> d) checking for each process instance whether a migration condition is met, and
> e) replacing the nodes of a original process definition in a running process instance satisfying the migration condition by corresponding nodes of a modified process definition such that said running process instance executes said modified process definition."

Examiner's Arguments

Regarding Claim 10, the Examiner asserted that:

"UNDERWOOD discloses a method for creating a process definition to be executed by a WFMS comprising the following steps:

> a) defining an original process definition to be executed in a work flow system {See Fig. 9A, Element 902};
> b) starting executing the process instance as per the original process definition {See Fig. 9A, Element 904};
> c) defining a modified process definition {See Fig. 9A, Element 906};
> d) checking for each process instance whether a migration condition is met {See Fig. 9A, Element 908}; and
> e) replacing the nodes of the original process definition in a running process instance satisfying the migration condition by the corresponding nodes of the modified process definition such that said running process

instance executes said modified process definition such that said running process instance executes said modified process definition {See Fig. 9A, Element 910}.

Specifically, the Examiner asserted that elements 902, 904, 906, 908 and 910 of Fig. 9A disclose elements a), b), c), d) and e) of Claim 10, respectively.

Appellant's Reply

Appellant submits that Fig. 9A of Underwood refers to a method for accessing services within a server without a need for knowledge of an application program interface of the server (col 41, lines 33-42), and has no relation to the method of creating a process definition to be executed by a WFMS according to Claim 10.

The creating of a role container (element 902, Fig.9A) is different from defining an original process definition to be executed in a workflow system as recited in the element a) of Claim 10. The role container is a concept in object-oriented programming referring to a collection of classes of objects, and defining such a container has no relation to defining the original process definition.

Defining a role class (element 904, Fig.9A) is different from starting execution of the process instance as per the original process definition as recited in element b) of Claim 10. The role class is a class being defined in the role container described above, and is not related to the starting of the execution of the process instance as per the original process definition.

Generating an attribute for the role class (element 906, Fig.9A) is different from defining a modified process definition as recited in element c) of Claim 10. The attribute is simply a characteristic of the role class, and does not have any relation to defining the modified process definition.

Making a role object in the role class with the attribute associated therewith (element 908, Fig.9A) is different from checking for each process instance whether a migration condition is met as recited in element d) of Claim 10. The role object is a particular instance of the role class having the characteristics of the attribute, and has nothing to do with checking whether a migration condition is met for each process instance.

Selecting a URL for the default start page attribute (element 910, Fig.9A) is different from replacing the nodes of an original process definition in a running process instance satisfying the migration condition by corresponding nodes of a modified process definition such that said running process instance executes said modified process definition as recited in element e) of Claim 10. As mentioned earlier, the attribute is the characteristic of the role class and selecting a URL for the attribute does not relate to replacing the nodes of the original process definition by corresponding nodes of the modified process definition.

As Underwood fails to disclose each and every element contained in Claim 10, Claim 10 is patentable over Underwood.

Furthermore, for reasons set forth above for Claim 1, Underwood does not disclose checking for each process instance whether a migration condition is met, and replacing the nodes of a original process definition in a running process instance satisfying the migration condition by corresponding nodes of a modified process definition such that said running process instance executes said modified process definition as recited in elements d) and e) of Claim 10. Thus, Claim 10 is patentable over Underwood at least for these reasons.

VIII. CONCLUSION

Reversal of the final rejection is respectfully requested and a Notice of Allowance is solicited.

Date: September 11, 2006

Mail Stop:
Hewlett-Packard Company
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

By: Thi Dang
Thi Dang
Reg. No.: 53,877

Hewlett-Packard Asia Pacific Pte.Ltd.
438A Alexandra Road
Blk A #02-12 Alexandra Technopark
Singapore 119967
Telephone No.: (65) 6374 6179
Facsimile No.: (65) 6270 0885

IX. CLAIMS APPENDIX

<u>The Appealed Claims</u>

1. (Previously presented) A method for executing a work flow in a WFMS having at least one process instance executing an original process definition, and migrating the said process instance to a changed process definition, said method comprising the following steps:

   a) checking each process instance during the execution of the original process definition whether the process instance meets a migration condition; and

   b) migrating each process instance during the execution of the original process definition to a modified process definition if the migration condition is met such that said process instance executes said changed process definition.

2. (Original) A method according to claim 1, wherein checking each process instance further comprises the following steps:

   - defining a set of worst case migration points (WMP), and
   - migrating the process instance to the modified process definition, if its execution has not gone beyond anyone of said worst case migration points (WMP).

3. (Original) A method according to claim 2, said step of defining a set of worst case migration points (WMP) comprises one of the following actions:

   - reading a set of worst case migration points (WMP) from an user input, or
   - computing a set of worst case migration points (WMP) based upon the original process definition and the modified process definition.

4. (Original) A method according to claim 3, said step of computing a set of worst case migration points (WMP) comprises the following steps:

- defining a set D including all nodes that are changed in the modified process definition with respect to the original process definition;
- determining a set P including all predecessor nodes for all nodes belonging to set D;
- determining a reachability matrix $R=(r_{ij})$ for all nodes belonging to set P, each row and column in the reachability matrix R representing a node in the order listed in P, wherein a node X representing a column is regarded as reachable from a another node Y representing a row, if there exists a path of arcs forward from X to Y; and
- determining the set of worst case migration points from the reachability matrix R.

5. (Original) A method according to claim 4, wherein the step of determining the reachability matrix $R=(r_{ij})$ further comprises the following actions:

- attributing a valued of x to each reachability matrix element r_{ij} if the predecessor node corresponding to said column j is reachable from the node corresponding to said row I;
- attributing a value of x to each reachability matrix element r_{zz}; and
- attributing a value of y to each reachability matrix element r_{ij} if the predecessor node corresponding to said column j is not reachable from the node corresponding to said row i.

6. (Original) A method according to claim 5, wherein the worst case migration points are determined by selecting those predecessor nodes for which the elements r_{ij} from the corresponding column add to a value of x.

7. (Original) A method according to claim 6, wherein a value of 1 is chosen for x and a value of 0 is chosen for y.

8. (Original) A method according to claim 1, wherein said step of checking each process instance during the execution of the original process definition whether it meets a migration condition further comprises of steps for checking whether the node(s) in the original process definition being currently executed is/are also present in the modified process definition.

9. (Original) A method according to claim 8, wherein the step of checking whether a node in the original process definition being currently executed is also present in the modified process definition is repeated upon executing of each node(s) of the original process definition until the migration of said process instance is completed.

10. (Previously presented) A method for creating a process definition to be executed by a WFMS comprising the following steps:

a) defining an original process definition to be executed in a work flow system;

b) starting execution of the process instance as per the original process definition;

c) defining a modified process definition;

d) checking for each process instance whether a migration condition is met;

e) replacing the nodes of the original process definition in a running process instance satisfying the migration condition by the corresponding nodes of the modified process definition such that said running process instance executes said modified process definition.

X. EVIDENCE APPENDIX

NONE

XI. RELATED PROCEEDINGS APPENDIX

NONE